



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/728,073	12/04/2000	Erik Hennum	07042-152001	3770

24852 7590 08/11/2006

INTERNATIONAL BUSINESS MACHINES CORP  
IP LAW  
555 BAILEY AVENUE , J46/G4  
SAN JOSE, CA 95141

EXAMINER

CHUONG, TRUC T

ART UNIT	PAPER NUMBER
----------	--------------

2179

DATE MAILED: 08/11/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

<b>Office Action Summary</b>	<b>Application No.</b>	<b>Applicant(s)</b>	
	09/728,073	HENNUM, ERIK	
	<b>Examiner</b>	<b>Art Unit</b>	
	Truc T. Chuong	2179	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

#### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

#### Status

- 1) ☒ Responsive to communication(s) filed on 19 January 2006.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

#### Disposition of Claims

- 4) ☒ Claim(s) 1-45 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-45 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

#### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 08 May 2001 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

#### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- \* See the attached detailed Office action for a list of the certified copies not received.

#### Attachment(s)

- |  |   |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892)                                   | 4) <input type="checkbox"/> Interview Summary (PTO-413)                     |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)               | Paper No(s)/Mail Date. _____  |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| Paper No(s)/Mail Date <u>11/09/05</u> <u>11-7-05</u>   | 6) <input type="checkbox"/> Other: _____                                    |

### **DETAILED ACTION**

This communication is responsive to a Petition, filed 01/19/06.

Claims 1-45 are pending in this application. Claims 1, 29, 30, 31, 32, and 45 are independent claims, and claims 1 and 30 are amended. This action is made final.

The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior office action.

### ***Claim Rejections - 35 USC § 103***

1. Claims 1-45 are rejected under 35 U.S.C. 103(a) as being unpatentable over FrontPage Screen Capture (herein after "FrontPage", Microsoft Copyright 1995-1999, figures 1-10) in view of Dinkelacker (U.S. Patent No. 6,092,068).

From Microsoft FrontPage → open a new file and type in "this is a test page" in Normal page (fig. 1) → select HTML page to view the source code (fig. 2) → from Help go to About Microsoft FrontPage (fig. 3) → Tech Support to view detail helps and instructions associated with the HTML code (figs. 3-4). There are an index section (keyword links) and a description section with links (annotation links) to view a particular help topic in details (figs. 4 and 6).

As to claim 1, FrontPage teaches a method, performed in a web-based environment on a computer system, of helping a user learn to implement an application, the method comprising:

presenting an annotation page that includes one or more annotations descriptive (figs. 4-6) of a source code file of a predetermined application (FrontPage Application to run HTML code), each annotation including keyword links, annotation links (an index section (keyword

Art Unit: 2179

links) and a description section with links (annotation links) to view a particular help topic in details (figs. 4 and 6)), detail of implementation of the application and explanation of code used in the application (the details of coding/application/other helps can be found by selecting the keyword links or the description section (annotation links) as shown in figs. 4-6);

providing a link to a resource in an annotation (fig. 6 shows annotation with links, and the links can be linked to other sources for more information about the code);

if the user selects a keyword link, presenting reference documentation associated with that keyword (from the index keywords of fig. 4 and 6, there will be a list of topics/links about the keyword will be displayed); and

if the user selects an annotation link, presenting another annotation descriptive of another source file of a predetermined application (fig. 6 shows if the user selects annotation links in the description section, the system will teach/describe other related coding topic in details to the user); however, FrontPage does not clearly show the side by side relationships between the program code and the index descriptions which can be automatically displayed without typing the keywords or manually searching from the user. Dinkelacker clearly shows that the tutor provides various information (tool tips) related the tag associated with mouse-over region in a particular section of the code in details to instruct the user with full description of that tag section (e.g., col. 3 lines 1-15, col. 4 lines 1-23, lines 53-65, and figs. 3-4). It would have been obvious at the time of the invention, a person with ordinary skill in the art would want to utilize the Tool Tip of Dinkelacker in the Coding Tech Support of FrontPage to prevent the confusion of the user/developer during the coding process to distinguish and minimize the choice(s) from the vast

Art Unit: 2179

information/instruction about a particular code/task (only a particular tag can be viewed in details, e.g., col. 5 lines 40-44).

As to claim 2, FrontPage teaches the method of claim 1 further comprising performing a predetermined application and presenting one or more annotations descriptive of the performed application in coordination with performance of the predetermined application (e.g., figs. 4 and 6).

As to claim 3, FrontPage teaches the method of claim 2 in which performing the predetermined application comprises receiving input from the user (e.g., coding of figs. 1-2).

As to claim 4, FrontPage teaches the method of claim 3 further comprising presenting another annotation page in coordination with performance of the predetermined application based on input from the user (e.g., figs. 4 and 6).

As to claim 5, FrontPage teaches the method of claim 4 in which presenting another annotation page comprises:

automatically and simultaneously calling an annotation request module including application, file, class and function names of a program unit for which detail should be displayed (figs. 4 and 6 show details of coding such as: header file, creating frames, customized tables, etc.);

mapping the request to an annotation (figs. 4 & 6); and informing a browser window in the web-based environment to display the other annotation page (Microsoft FrontPage is a web-based environment).

As to claim 6, FrontPage teaches the method of claim 3 in which another annotation page is presented in coordination with performance of the predetermined application (figs. 1-6).

As to claim 7, FrontPage teaches the method of claim 6 further comprising automatically generating a global table (the listing topic shows in figs. 4 & 6) of contents comprising links to annotations by parsing structured links in web pages including annotation pages (figs. 4 & 6).

As to claim 8, FrontPage teaches the method of claim 7 in which the links in the global table of contents are synchronized with presented annotations by highlighting links corresponding to a current annotation page (highlighting, fig. 6).

As to claim 9, FrontPage teaches the method of claim 8 in which the global table of contents is presented in a first frame of a first browser window (e.g., fig. 7), the annotation page is presented in a second frame of the first browser window, and the predetermined application is performed in a second browser window (e.g., fig. 7 shows the application is performed in a second frame).

As to claim 10, FrontPage teaches the method of claim 2 in which performing the predetermined application comprises launching a Java applet or application (e.g., fig. 8 shows Java Applet).

As to claim 11, FrontPage teaches the method of claim 10 in which launching the Java applet or application comprises calling a Java application programming interface to ask a web browser to show the annotation page (e.g., figs. 6-8).

As to claim 12, FrontPage teaches the method of claim 2 in which performing the predetermined application comprises downloading a hyper-text markup language page containing a Java applet (e.g., fig. 9 shows a help link to open a web browser for downloading java related products).

As to claim 15, FrontPage teaches the method of claim 2 in which the annotation page is presented in a first browser window and the predetermined application is performed in a second browser window (e.g., figs. 6-9).

As to claim 16, FrontPage teaches the method of claim 1 in which application implementation detail includes text descriptive of the application, fragments of source code from the application, or both (e.g., figs. 4-8).

As to claim 17, FrontPage teaches the method of claim 16 in which source code fragments are imported directly from the source code file of the presented application (e.g., figs. 7-9).

As to claim 18, FrontPage teaches the method of claim 1 further comprising automatically generating the annotation page descriptive of the source code file of a predetermined application (e.g., figs. 4-7).

As to claim 19, FrontPage teaches the method of claim 18 in which generating the annotation page comprises:

receiving a source code file that has embedded text marked up with instructions (e.g., fig. 6);

parsing the source code to determine a structure of the predetermined application (e.g., figs. 6-9); and

generating one or more annotations based on the predetermined application structure and instructions (e.g., figs. 6-9).

As to claim 20, FrontPage teaches the method of claim 19 in which generating the annotation page comprises:

Art Unit: 2179

generating one or more annotation links for navigating the annotations of the predetermined application (e.g., figs. 8-9);

generating application implementation detail based on the embedded information (e.g., figs. 8-9); and

generating one or more keyword links for reference documentation (e.g., figs. 6-9).

As to claim 21, FrontPage teaches the method of claim 20 in which generating the annotation page comprises highlighting the keyword links and the annotation links in the annotation page (e.g., figs. 6-9).

As to claim 22, FrontPage in view of Dinkelacker teaches the method of claim 19 further comprising automatically updating the annotation page descriptive of the source code file of the predetermined application when an updated source code file is received (e.g., FrontPage shows the download information will be updated, and Dinkelaker also teaches that the system is a dynamic tutoring system, col. 3 lines 1-3).

As to claim 23, FrontPage teaches the method of claim 1 further comprising automatically generating a global table of contents by parsing the plurality of annotations for annotation links (e.g., figs. 4-8).

As to claim 24, FrontPage teaches the method of claim 23 further comprising providing the global table of contents, in which the global table of contents comprises links to annotations (e.g., figs. 4-9).

As to claim 25, FrontPage teaches the method of claim 23 further comprising generating a local table of contents, in which the local table of contents comprises links to web pages including annotation pages relating to an application (e.g., figs. 8-9).



As to claim 26, FrontPage teaches the method of claim 25 further comprising providing the local table of contents when a local link in the global table of contents is selected (e.g., figs. 4-9).

As to claim 27, FrontPage teaches the method of claim 1 in which the presented annotation page is descriptive of the performed application and the annotation page is presented in coordination with performance of the predetermined application (e.g., fig. 9).

As to claim 28, FrontPage teaches the method of claim 1 further comprising:  
generating a source code file stripped of annotation mark up, the generated source code file including source code of the application but not including text from the annotations (the different annotation link having different information about the code without the comment in the code, e.g., fig. 10);

presenting the stripped source code file (e.g., fig. 10); and

permitting the user to edit the stripped source code file (the source code from fig. 8 can be copied/pasted/modified/saved as a file).

As to claim 29, it is individually similar in scope to claim 1 above; therefore, rejected under similar rationale.

As to claim 30, FrontPage teaches a method, performed in a web-based environment on a computer system, for teaching a user to implement an application, the method comprising:

automatically assembling a global table of contents based on content in the environment, the global table of contents including a plurality of links to content within the environment (FrontPage teaches global table of contents, which is the first frames with keywords/index with links shown fig. 4-7);

Art Unit: 2179

providing the global table of contents (e.g., figs. 4-7);

generating a local table of contents that includes links to content that orient the user within a local topic (a second frame shows the annotations to view a particular help topic in details, e.g., fig. 7 shows listing of other links in the annotation section) and that provides a structured hierarchical view at a local corner of an unstructured web page of links (figs. 5-6 show hierarchical structure of links at the corner of the help page); and

permitting the user to select links from the local table of contents to access local topics (e.g., figs. 7-10).

As to claim 31, FrontPage teaches a method, performed in a web-based environment on a computer system, of teaching a user to implement an application, the method comprising:

providing a plurality of predefined interactive examples; and performing one or more of the predefined interactive examples in response to user selection (e.g., figs. 8-10);

presenting one or more annotations descriptive of the performed interactive example in coordination with performance of the predefined interactive example (e.g., fig. 10); and

allowing the user to selectively explore different aspects of the performed interactive example, the annotations, or both (e.g., figs. 7-10).

As to claim 32, this is a system claim of method claims 1 and 31. Note the rejections of claims 1 and 31 above.

As to claim 33, FrontPage teaches the system of claim 32 further comprising a utility through which the user can access source code associated with a predefined interactive application (e.g., fig. 10).

Art Unit: 2179

As to claim 34, FrontPage teaches the system of claim 33 in which the utility enables the user to view or copy a predefined interactive application's source code (e.g., the source code from fig. 8 can be copied/pasted/modified/saved as a file).

As to claim 35, this is a system claim of method claim 16. Note the rejection of claim 16 above.

As to claim 36, this is a system claim of method claim 20. Note the rejection of claim 20 above.

As to claim 37, FrontPage teaches the system of claim 32 further comprising a web-browser window that includes a framework that comprises:

a content frame that displays the annotations; a framework applet that displays a navigation bar; and a table of contents frame that displays a table of contents hierarchy of links (figs. 4-10 shows navigation bar).

As to claim 38, this is a system of method claim 10. Note the rejection of claim 10 above.

As to claim 39, FrontPage in view of Dinkelacker teaches the system of claim 37 in which a Java Script automatically determines whether the framework is present in the web browser window (Dinkelacker shows using the JavaScript to tag the section, e.g., col. 3 lines 40-50 and fig. 3), and if the framework is present, notifies the framework applet about the content in the framework (e.g., figs. 3-4).

As to claim 40, FrontPage teaches the system of claim 39 in which the table of contents automatically highlights a link in the hierarchy based on the content in the framework (e.g., figs. 5, 7, 9, and 10).

As to claim 41, FrontPage teaches the system of claim 40 in which the user accesses an annotation page by selecting a link in the table of content's hierarchy (e.g., fig. 5).

As to claim 42, FrontPage teaches the system of claim 40 in which the user accesses an annotation page by interacting with the navigation bar (e.g., figs. 4-7).

As to claim 43, FrontPage teaches the system of claim 40 in which the table of contents highlights the hierarchy based on an annotation page displayed in the content frame (e.g., fig. 5).

As to claim 44, FrontPage teaches the system of claim 37 in which the table of contents is dismissible or resizable (e.g., figs. 4-10 show the section can be closed or minimized).

As to claim 45, FrontPage teaches a web-based computer system for teaching a user to implement an application, the system comprising:

a web-browser window that includes a content frame, a framework applet, and a table of contents frame that displays a global table of contents hierarchy of links related to content in the content frame (e.g., figs. 5-10);

one or more annotations displayed in the content frame, each annotation describing a predefined interactive application and including links to other content (e.g., figs. 5-10); and

a table of contents window that displays a local table of contents hierarchy of links related to local content in the displayed annotation (figs. 5, 8, and 10).

### ***Response to Arguments***

2. Applicant's arguments filed 11/09/05 have been fully considered but they are not persuasive.

Applicants argued and Examiner disagrees with the followings:

- a. *Filing date of the current application, filed December 04, 2000 (it can go back to December 17, 1999) is prior to Microsoft FrontPage and Dinkelaker (U.S. Patent No. 6,092,068), and the applicant believes that it can go back to July 7, 1997 because of the Terminal Disclaimer between the current application and Hennum et al. (U.S. Patent No. 6,724,401 B1).*

Based on the Terminal Disclaimer (filed on December 27, 2004), which means the current application 09/728,073 and Hennum et al. (U.S. Patent No. 6,724,401 B1) is also a division of application No. 08/888,925 (Now Patent No. 6,259,445), filed on July 7, 1997 are now disclaimed. It is not true because the application No. 08/888,925 (Now Patent No. 6,259,445), filed on July 7, 1997, is the division of application, which means that the current application and the divisional application might not cover the same features of the inventions; therefore, the current application can only go back to December 17, 1999 unless the applicant clearly explains the features of both cases.

- b. *FrontPage does not teach that the markup language assists the programmer by showing example annotation relating to how to write functional code.*

The programmer uses Microsoft FrontPage Application to code/run HTML code. If the programmer has questions during coding/running, he/she will have answer by selecting Tech Support to view detail helps and instructions associated with the HTML code (figs. 3-4). There are also an index section (keyword links) and a description section with links (annotation links) to view a particular help topic in details (figs. 4 and 6).

- c. *FrontPage cannot demonstrate user interaction with source code and annotation.*

The programmer can select/view/ask a particular help topic or keywords of the program code during coding, then the annotations view the particular help topic in details to help the programmer to understand the concept of that program code as shown in figs. 4-6.

- d. *Claim 29 cannot be rejected under a similar rationale as claim 1.*

Based on the claim language of both claims 1 and 29, they are covered the same limitations such as predetermined application, annotation page, explanation of code, and links to annotations. The only different is claim 29 is broader than claim 1; therefore, claim 29 can be rejected under a similar rationale as explained in claim 1 above.

- e. *FrontPage does not teach the global table and local table.*

FrontPage teaches global table of contents, which is the first frames with keywords/index with links shown fig. 4-7, and local table/second frame shows the annotations to view a particular help topic in details (e.g., fig. 7 shows listing of other links in the annotation section).

- f. *There is no motivation or suggestion to combine Microsoft FrontPage and Dinkelacker.*

In response to applicant's argument that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found

Art Unit: 2179

either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988) and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, it is well known, or it would have been obvious at the time of the invention to utilize the Tool Tip of Dinkelacker in the Coding Tech Support of Microsoft FrontPage to prevent the confusion of the user/developer/programmer during the coding process to distinguish and minimize the choice(s) from the vast information/instruction about a particular code/task (Dinkelacker, only a particular tag can be viewed in details, e.g., col. 5 lines 40-44).

### ***Conclusion***

3. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Art Unit: 2179

4. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Truc T. Chuong whose telephone number is 571-272-4134. The examiner can normally be reached on M-Th and alternate Fridays 8:30 AM - 5:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Weilun Lo can be reached on (571) 272-4847. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Truc T. Chuong

08/01/06

  
BA HUYNH  
PRIMARY EXAMINER